σ	42	ω	2	Ĺ	
	Н	U	Н	6	Hits
(transaction near6 lock\$3) and ((savepoint or save?point or checkpoint or syncpoint or syncpoint or syncpoint or sync?point) near6 lock\$3) and (rollback or roll?back or (roll\$3 adj back) or undo)	(((("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (savepoint or save?point or checkpoint or check?point or sync?point or sync?point)) and ((("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (rollback or roll?back or (roll\$3 adj back) or undo))	"6275832" 185699" " 897641"). roll?bac ck) or un	(("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (savepoint or save?point or checkpoint or check?point or syncpoint or sync?point)	("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.	Search Text
USPAT; US-PGPUB; EPO; JPO; IBM_TDB	USPAT; US-PGPUB	USPAT; US-PGPUB	USPAT; US-PGPUB	USPAT; US-PGPUB	DBs
S-PGPUB; 2003/07/10 ; IBM_TDB18:15	2003/07/10 18:08	2003/07/10 18:00	2003/07/10 18:00	2003/07/10 17:59	Time Stamp

φ	ω	7	σ	
43	356	327	7 5	Hits
((transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point or "check point" or ((subtransaction or steps!) near? (point or portion or snapshot or snap?shot)))) and ((transaction with lock\$3) and ((savepoint or save?point or "save point" or checkpoint or check?point or subtransaction or sub?transaction or sub?transaction or sub?transaction or sub?transaction or steps!) wi	transaction with lock\$3) and (savepoint or save?point or save point or heck?point or "check point" or ubtransaction or steps!) with	nsaction with (savepoi ?point or "save point" kpoint or check?point ck point" or btransaction or transaction or steps!)	((transaction near6 lock\$3) and ((savepoint or save?point or checkpoint or check?point or syncpoint or sync?point or syncpoint or sync?point) near6 lock\$3) and (rollback or roll?back or (roll\$3 adj back) or undo)) and @ad<=20010628	Search Text
USPAT; US-PGPUB; EPO; JPO; IBM_TDB	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	USPAT; U EPO; JPO	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	DBs
2003/07/10	2003/07/10	S-PGPUB; 2003/07/10 ; IBM_TDB19:10	PGPUB; 2003/07/10 IBM_TDB19:05	Time Stamp

10	
21	Hits
(((transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point or "check point or "((subtransaction or steps!) near? (point or portion or steps!) near? (point or portion or snapshot or snap?shot)))) and ((transaction with lock\$3) and ((savepoint or checkpoint or sub?transaction or sub?transaction or sub?transaction or sub?transaction or undo or revers\$3) near5 (savepoint or save?point or check?point or save?point or save?point or save?point or check?point or save?point or sub?transaction or sub?tr	Search Text
USPAT; EPO; JP	DBs
US-PGPUB; 2003/07/10 O; IBM_TDB 19:13	Time Stamp

70 70 O = - O H S O	1 1 1 9	Hits
d ((rollba l\$3 adj ba \$3) near5 e?point or heckpoint heck point r steps!))	kpoint or check?point or ck point or check?point or ck point or check?point or ck point or or steps!) near2 cransaction or snapshot or ?shot)))) and ((transaction lock\$3) and ((savepoint or ?point or "save point" or kpoint or check?point or subtransaction charaction or steps!)	on with (s
	USPAT; US- EPO; JPO;	DBs
	PGPUB; 2003/07/10 IBM_TDB19:50	Time Stamp

18 3	17 2	16 91	15 9	14 2	
309	2749			Φ	Hits
((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))	(707/200 707/201 707/202 707/8).ccls.	or kout or out)) point o point o as\$3 u?lock	((rollback or roll?back or (roll\$3 adj back) or backout or back?out or (back\$3 adj out)) with (savepoint or save?point or save point")) and (releas\$3 near2 (lock or x?lock or u?lock or s?lock or latch))	<pre>(rollback or roll?back or (roll\$3 adj back) or backout or back?out or (back\$3 adj out)) with (savepoint or save?point or "save point")</pre>	Search Text
USPAT; US-PGPUB; EPO; JPO; IBM_TDB	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	rUSPAT; US-PGPUB; EPO; JPO; IBM_TDB	USPAT; US- EPO; JPO;	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	DBs
PGPUB; 2003/07/10 IBM_TDB20:05	UB; 2003/07/10 L_TDB20:04	PGPUB; 2003/07/10 IBM_TDB20:04	PGPUB; 2003/07/10 IBM_TDB19:59	-PGPUB; 2003/07/10 IBM_TDB19:58	Time Stamp

	19 48	
H	ω	Hits
(((((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or checkpoint or check?point or check?point or "check point")) and ((rollback or roll?back or (roll\$3 adj back) or backout or back?out or (back\$3 adj out)) with (steps! or savepoint or save?point or check?point or checkpoint or check?point or "check point"))) and (releas\$3 with (lock or	(((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or checkpoint or "save point" or "checkpoint or check?point or "check point")) and ((rollback or roll?back or (roll\$3 adj back) or backout or back?out or (back\$3 adj out)) with (steps! or savepoint or save?point or check?point or check?point or "check?point"))	Search Text
USPAT; US-PGPUB; 2003/07/10 EPO; JPO; IBM_TDB20:07	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	DBs
2003/07/10	S-PGPUB; 2003/07/10; IBM_TDB20:07	Time Stamp

	Hits	Search Text	DBs	Time Stamp
N H	19	((((((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or checkpoint or "save point" or checkpoint or check?point or check?point or (rollback or roll?back or (roll\$3 adj back) or backout or back?out or (back\$3 adj out)) with (steps! or savepoint or save?point or check?point or "checkpoint or and (releas\$3 with (lock or x?lock or u?lock or s?lock or latch))) and @ad<=20010628	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	PGPUB; 2003/07/10 IBM_TDB20:07
22		30124" "5452 5025" "60472 6075" "57376	USPAT; US-PGPUB	2003/07/11 11:14
23	T	((rollback or roll?back or (roll\$3 adj back)) near4 (savepoint or save?point or "save point" or intermediate or partial)) same ((releas\$3 near1 lock) or unlock\$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	PGPUB; 2003/07/11 IBM_TDB 11:16
24	264	(indrajit bertino jajodia mancini).in.	USPAT; US-PGPUB	2003/07/11 11:52
25	0	((indrajit bertino jajodia mancini).in.) and commit	USPAT; US-PGPUB	2003/07/11 11:52
26	7	((indrajit bertino jajodia mancini).in.) and transaction	USPAT; US-PGPUB	2003/07/11 11:53
27	28	(watts and julie).in.	USPAT; US-PGPUB; 2004 EPO; JPO; IBM_TDB14:1	2004/02/23 14:18

	Hits	Search Text d julie).in.) a	a tarad
28 12	2 ((watts and julie).in.) savepoint or save?point save?point	USPAT; US-PGPUB; EPO; JPO; IBM_TD
29 2	аркат ((database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll\$3 adj back)) and (lock\$3 with (savepoint or save?point))	USPAT; U EPO; JPO
30 46		database or data?base) and ransaction and (savepoint or ave?point or "save point") and rollback or roll?back or roll\$3 adj back)	USPAT; US EPO; JPO,
3 1 4 4	р к й ц ~ ~ и ц ~	((database or data?base) and transaction and (savepoint or save?point or "save point") and (rollback or roll?back or (roll\$3 adj back))) not ((database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll\$3 adj back)) and (lock\$3 with (savepoint or save?point)	USPAT; US EPO; JPO;

33 7		32 11	Hits
ommit or committed or mmitting)))	((savepoint or save?point or "save point" or (partial adj (rollback or roll?back))) same lock\$3) and transaction and [(repeatable adj read) or ((readE) or maintain) with lock\$3 with	((((database or data?base) and transaction and (savepoint or save?point or "save point") and (rollback or roll?back or (roll\$3 adj back))) not ((database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll\$3 adj back)) and (lock\$3 with (savepoint or save?point))) and (lock\$3 with (maintain\$3 or commit or committing or committed))	Search Text
	SPAT; US-PGPUB;	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	DBs
2004/02/24 B13:20		2004/02/23	Time Stamp

36 15	Hits
(((707/200 707/201 707/202 707/8).ccls.) and (savepoint or save?point or "save point" or (partial adj (rollback or roll?back))) and ((repeatable adj read\$3) or (read\$3 same lock\$3 same (commit or committed)	Search Text
USPAT; US-PGPUB; EPO; JPO; IBM_TDB	DBs
UB; 2004/02/24 _TDB13:23	Time Stamp



> home | > about | > feedback | > log

US Patent & Trademark Office



Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: [(((savepoint or save?point or "save point" or (partial <near/1> (rollback or roll?back))) <sentence> lock*) and (transaction <paragraph> lock*) and ((repeatable <near/1> read*) or (lock* <sentence> (read or maintain* or duration) <sentence> (commit or committing or committed))))<AND>(meta_published_date <= 06-01-2001)

Found 16 of 127,132 searched.

Search within Results

> Advan	ced Sea	rch > Searc	h Help/Tips		
•			Publication Date	● Binder	
Results 1	- 16 of	`16 <u>short</u> l			

1 An advanced commit protocol for MLS distributed database systems 100%

Indrajit Ray, Elisa Bertino, Sushil Jajodia, Luigi Mancini
Proceedings of the 3rd ACM conference on Computer and communications security January 1996

2 Analysis of locking behavior in three real database systems

100%

Vigyan Singhal, Alan Jay Smith

The VLDB Journal — The International Journal on Very Large Data Bases February 1997

Volume 6 Issue 1

Concurrency control is essential to the correct functioning of a database due to the need for correct, reproducible results. For this

reason, and because concurrency control is a well-formulated problem, there has developed an enormous body of literature studying the performance of concurrency control algorithms. Most of this literature uses either analytic modeling or random number-driven simulation, and explicitly or implicitly makes certain assumptions about the behavior of transactions and t ...

3 Ordered shared locks for real-time databases

100%

Divyakant Agrawal, Amr El Abbadi, Richard Jeffers, Lijing Lin The VLDB Journal — The International Journal on Very Large Data Bases January 1995

Volume 4 Issue 1

We propose locking protocols for real-time databases. Our approach has two main motivations: First, locking protocols are widely accepted and used in most database systems. Second, in real-time databases it has been shown that the blocking behavior of transactions in locking protocols results in performance degradation. We use a new relationship between locks called ordered sharing to eliminate blocking that arises in the traditional locking protocols. Ordered sharing eliminates blocking of read ...

4 Efficient concurrency control in multidimensional access methods

100%

Maushik Chakrabarti, Sharad Mehrotra

ACM SIGMOD Record, Proceedings of the 1999 ACM SIGMOD international conference on Management of data June 1999

Volume 28 Issue 2

The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a " commercial-strength" database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

5 Concurrency control: methods, performance, and analysis

100%

Alexander Thomasian

ACM Computing Surveys (CSUR) March 1998 Volume 30 Issue 1

6 Concurrency and recovery in generalized search trees

100%

Marcel Kornacker, C. Mohan, Joseph M. Hellerstein

ACM SIGMOD Record, Proceedings of the 1997 ACM

SIGMOD international conference on Management of data June
1997

Volume 26 Issue 2

This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

7 Recovery protocols for shared memory database systems

100%

Lory D. Molesky, Krithi Ramamritham

ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data May 1995

Volume 24 Issue 2

Significant performance advantages can be gained by implementing a database system on a cache-coherent shared memory multiprocessor. However, problems arise when failures occur. A single node (where a *node* refers to a processor/memory pair) crash may require a reboot of the entire shared memory system. Fortunately, shared memory multiprocessors that isolate individual node failures are currently being developed. Even with these, because of the side effects of the cache coherency protocol, ...

<u>8 Long-duration transaction support in design databases</u>

100%

Maldemar Wieczerzycki

Proceedings of the fourth international conference on Information and knowledge management December 1995

9 Altruistic locking

100%

Kenneth Salem, Héctor García-Molina, Jeannie Shands

ACM Transactions on Database Systems (TODS) March 1994

Volume 19 Issue 1

Long-lived transactions (LLTs) hold on to database resources for relatively long periods of time, significantly delaying the completion of shorter and more common transactions. To alleviate this problem we propose an extension to two-phase locking, called altruistic locking, whereby LLTs can release their locks early. Transactions that access this released data are said to run in the wake of the LLT and must follow special locking rules. Like two-phase locking, altruistic locking is easy to ...

10 IBM's relational DBMS products: features and technologies

100%

C. Mohan

ACM SIGMOD Record, Proceedings of the 1993 ACM SIGMOD international conference on Management of data June 1993

Volume 22 Issue 2

This paper very briefly summarizes the features and technologies implemented in the IBM relational DBMS products. The topics covered include record and index management, concurrency control and recovery methods, commit protocols, query optimization and execution techniques, high availability and support for parallelism and distributed data. Some indications of likely future product directions are also given.

11 Reducing recovery constraints on locking based protocols

100%

G. Alonso, D. Agrawal, A. El Abbadi

Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems May 1994
Serializability is the standard correctness criterion for concurrency control. To ensure correctness in the presence of failures, recoverability is also imposed. Pragmatic considerations result in

further constraints, for instance, the existing log-based recovery implementations that use before-images warrant that transaction executions be strict. Strict executions are restrictive, thus sacrificing concurrency and throughput. In this paper we identify the relation between the recovery mecha ...

12 ARIES/IM: an efficient and high concurrency index management

100%

method using write-ahead logging

C. Mohan, Frank Levine

ACM SIGMOD Record, Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

This paper provides a comprehensive treatment of index management in transaction systems. We present a method, called ARIESIIM (Algorithm for Recovery and Isolation Exploiting Semantics for Index Management), for concurrency control and recovery of B+-trees. ARIES/IM guarantees serializability and uses write-ahead logging for recovery. It supports very high concurrency and good performance by (1) treating as the lock of a key the same lock as the one on the ...

13 Efficient and flexible methods for transient versioning of records to

100%

avoid locking by read-only transactions

C. Mohan, Hamid Pirahesh, Raymond Lorie

ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

We present efficient and flexible methods which permit read-only transactions that do not mind reading a possibly slightly old, but still consistent, version of the data base to execute without acquiring locks. This approach avoids the undesirable interferences between such queries and the typically shorter update transactions that cause unnecessary and costly delays. Indexed access by such queries is also supported, unlike by the earlier methods. Old versions of records are maintained only ...

14 Using delayed commitment in locking protocols for real-time

100%

databases

D. Agrawal, A. El Abbadi, R. Jeffers

ACM SIGMOD Record, Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

In this paper, we propose locking protocols that are useful for real-time databases. Our approach is motivated from two main observations. First, locking protocols are widely accepted and used in most database systems. Second, in real-time databases it has been shown that the blocking behavior of transactions in locking protocols results in performance degradation. We use a new relationship between locks called ordered sharing to eliminate blocking that arises in the traditional locking pro ...

15 ARIES: a transaction recovery method supporting fine-granularity

100%

d locking and partial rollbacks using write-ahead logging

C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz

ACM Transactions on Database Systems (TODS) March 1992 Volume 17 Issue 1

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

16 An approach to eliminate transaction blocking in locking protocols

D. Agrawal, A. El Abbadi, R. Jeffers

Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems July 1992

100%

Results 1 - 16 of 16 short listing

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.



> home | > about | > feedback | > log

US Patent & Trademark Office



Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: [((savepoint or save?point or save point or (partial <near/1> (rollback or roll?back))) and transaction and ((repeatable <near/1> read) or (lock\$3 <sentence> (read or maintain* or duration) <sentence> (commit or committed or committing))))<AND>(meta_published_date <= 06-01-2001)]
Found 9 of 127,132 searched.

a	* . 1 *	-	4	
Search	within	ĸ	ACII	ltc
Scarcii	WILLIE	-17	Cou	เนอ

> Advanced Search > Search Help/Tips						
*************************************				**************************	***************************************	
Sort by:			Publication Date		⊗ Binder	
Results 1 - 9 of 9 short listing						

1 Failure isolation and recovery in composite multidatabases

100%

Dexter P. Bradshaw

Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research October 1994

Most concurrency control schemes for guaranteeing global serializability in composite multidatabase systems are susceptible to rollbacks. Conservative schemes generate rollbacks because of transaction timeouts, while those of optimistic schemes are caused by certification failures. Typically, rollbacks on any branch of a flat distributed transaction cause a global abort. Global aborts during multidatabase composition degrade performance because of a waste of resources and reductions in multidata ...

2 Concurrency control: methods, performance, and analysis

100%

Alexander Thomasian

ACM Computing Surveys (CSUR) March 1998 Volume 30 Issue 1

3 ARIES/IM: an efficient and high concurrency index management

100%

method using write-ahead logging

C. Mohan, Frank Levine

ACM SIGMOD Record, Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

This paper provides a comprehensive treatment of index management in transaction systems. We present a method, called ARIESIIM (Algorithm for Recovery and Isolation Exploiting Semantics for Index Management), for concurrency control and recovery of B+-trees. ARIES/IM guarantees serializability and uses write-ahead logging for recovery. It supports very high concurrency and good performance by (1) treating as the lock of a key the same lock as the one on the ...

4 ARIES: a transaction recovery method supporting fine-granularity

100%

locking and partial rollbacks using write-ahead logging

C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz

ACM Transactions on Database Systems (TODS) March 1992 Volume 17 Issue 1

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

5 An approach to eliminate transaction blocking in locking protocols

D. Agrawal, A. El Abbadi, R. Jeffers

100%

Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems July 1992

6 Efficient concurrency control in multidimensional access methods

100%

Kaushik Chakrabarti, Sharad Mehrotra

ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international conference on Management of data June 1999

Volume 28 Issue 2

The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a "commercial-strength" database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

7 Concurrency and recovery in generalized search trees

100%

Marcel Kornacker, C. Mohan, Joseph M. Hellerstein

ACM SIGMOD Record, Proceedings of the 1997 ACM

SIGMOD international conference on Management of data June
1997

Volume 26 Issue 2

This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

8 IBM's relational DBMS products: features and technologies

99%

C. Mohan

ACM SIGMOD Record, Proceedings of the 1993 ACM SIGMOD international conference on Management of data June

1993

Volume 22 Issue 2

This paper very briefly summarizes the features and technologies implemented in the IBM relational DBMS products. The topics covered include record and index management, concurrency control and recovery methods, commit protocols, query optimization and execution techniques, high availability and support for parallelism and distributed data. Some indications of likely future product directions are also given.

9 DLFM: a transactional resource manager

80%

Hui-I Hsiao, Inderpal Narang

ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data May 2000

Volume 29 Issue 2

The DataLinks technology developed at IBM Almaden Research Center and now available in DB2 UDB 5.2 introduces a new data type called DATALINK for a database to reference and manage files stored external to the database. An external file is put under a database control by "linking" the file to the database. Control to a file can also be removed by "unlinking" it. The technology provides transactional semantics with respect to linking or unlinking the file when DATALINK ...

Results 1 - 9 of 9 short listing

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.